

SoC Linux 道場【其ノ五】

Linux カーネルの入手とコンパイル、 カスタム・ドライバの作成とコンパイル(その1)

Ver.13.1



2017年8月 Rev.2

ELSENA,Inc.





SoC Linux 道場【其ノ五】

Linux カーネルの入手とコンパイル、カスタム・ドライバの作成とコンパイル(その1)

<u>目次</u>

1. <u>はじめに</u>	
2. Linux カーネルの入手とコンパイル	4
2-1. インターネットに接続するためのネットワーク設定の変更	4
2-1-1. Windows のネットワーク設定	5
2-1-2. Vine Linux のネットワーク設定	7
2-1-3. VMware のネットワーク設定	11
2-1-4. インターネットへの接続確認	12
2-2. Linux カーネルの入手	
2-3. Linux カーネルのコンパイル	16
2-4. Helio に接続するためのネットワーク設定の変更	19
2-4-1. Windows のネットワーク設定	19
2-4-2. Vine Linux のネットワーク設定	21
2-4-3. VMware のネットワーク設定	25
2-4-4. Helio との接続確認	26
2-5. コンパイルした Linux カーネルに差し換えて Helio を起動してみる	
2-5-1. Windows 側に zImage ファイルをコピーする	27
2-5-2. microSD カード内の zImage ファイルを差し換える	29
2-5-3. zImage ファイルを差し換えた microSD カードで Helio を起動する	
3. <u>カスタム・ドライバの作成とコンパイル(その1)</u>	
改版履歴	



1. <u>はじめに</u>

今回は、アルテラ SoC FPGA 向け Yocto ソース・パッケージを利用した Linux[®] カーネルの入手と⊐ ンパイルの手順について解説します。

RocketBoards.org ポータル・サイトから Linux カーネルの git ツリーのクローンを作成し、その後、カーネル・コンフィギュレーションとコンパイルを実行します。

生成物として zImage という Linux カーネルのバイナリ・イメージができます。

この zImage ファイルを本連載の <u>【其ノ壱】</u> "ビルド済み SD カード・イメージを使用した Helio ボードでの Linux ブートの確認" で作成した microSD カード内にある zImage ファイルと差し換えることで、今回コンパイルした Linux カーネルの起動を Helio ボードで確認することができます。

また、「カスタム・ドライバの作成とコンパイル(その1)」として、簡単な Hello ドライバ(Hello メッセージ を表示するドライバ)をコンパイルして Helio ボードに NFS サーバー経由で転送してから実行してみます。

尚、この資料の説明で使用している主な開発環境は以下の通りです。

項番	項目	内容
1	ホスト OS	Microsoft Windows 7 Professional sp1 日本語版(64 bit)
2	ゲスト OS	Vine Linux 6.5 x86_64 この資料では、Linux 開発環境として、Vine Linux ディストリビューションを使用します。 詳細については、 <u>SoC Linux 道場【其ノ弐】</u> を参照ください。
3	仮想 OS 実行 環境	OS を仮想的に実行するための環境です。 この説明では、「VMware Player for Windows」と呼ばれるフリーウェア・ソフトを使用しています。 詳細については、 <u>SoC Linux 道場【其ノ弐】</u> を参照ください。
4	クロス・コ ンパイラ	ターゲット(Helio)向けの実行イメージを生成するためのコンパイラです。 この説明では、Linaro から提供されている 32-bit ARMv7 Cortex-A 向け Linux GNU クロス・ツー ルチェーンを使用しています。 詳細については、 <u>SoC Linux 道場【其ノ参】</u> を参照ください。
5	Linux カーネル ソース・コ ード	 アルテラ SoC FPGA 向け Linux カーネル・ソース・コード RocketBoards.org ポータル・サイトの以下のページに行くと、アルテラ SoC FPGA 向けの各 バージョン/リリースの Linux カーネル・ソース・コードがあります。 この説明では、 rel_socfpga-3.9-rel_14.02.02 というバージョン/リリースを使用しています。 ■ アルテラ SoC FPGA 向けの各バージョン/リリースの Linux カーネルのページ https://github.com/altera-opensource/linux-socfpga ■ 詳細は、以下の「Linux - Getting Started Using Git Trees」を参照してください。 http://www.rocketboards.org/foswiki/Documentation/GitGettingSTarted

【表 1.1】この資料の説明で使用している主な開発環境

	MA	SoC Linux 道場【其ノ五】 Linux カーネルの入手とコンパイル、カスタム・ドライバの作成とコンパイル(その1) ELSENA
6	Helio ボード	動作確認でターゲット・ボードとして使用する、アルテラ Cyclone [®] V SoC を搭載した マクニカ Helio ボードです。
		Helio には複数のリビジョンが存在しますが、この資料では、Rev1.2 または Rev1.3 を使用して動作 確認を行っています。
		■ Helio ボード Rev1.2 http://www.rocketboards.org/foswiki/Documentation/HelioResourcesForRev12
		■ Helio ボード Rev1.3 http://www.rocketboards.org/foswiki/Documentation/HelioResourcesForRev13

2. Linux カーネルの入手とコンパイル

本章はご自分の環境で、Helio(アルテラ SoC FPGA)用のカーネルを入手する際の参考にしてください。

2-1. インターネットに接続するためのネットワーク設定の変更

この連載において、ここまでのネットワーク設定と接続としては、ホスト PC と Helio を 1対1 で接続する 閉じた環境を構成していました。

git ツリーのクローンを作成し、Linux カーネルをコンパイルする場合、外部のインターネットと接続するので、ネットワーク設定を変更する必要があります。



【図 2-1.1】 ホスト PC をインターネットに接続する

会社内の LAN をご使用の場合は、プロキシの設定などがある場合もありますので、社内のネットワーク 管理者に確認してください。

ここでの説明は、基本的な設定として参考にしてください。

【注記】

<u>SoC Linux</u> 道場 【其ノ弐】 で、Helio の IP アドレスを付与するために、Vine Linux 上で「端末」を 開いて shell が起動すると、DHCP サーバーが起動するように設定しました。

今回の【<u>其ノ五</u>】では、外部のインターネットに接続するため、DHCP サーバーが動作しているとネットワーク上問題が発生することも考えられます。

【<u>其ノ弐】</u>で、DHCP サーバーが起動するように設定している場合は、外部のインターネットに接続 する前に、DHCP サーバーを停止させてください。

4/36

OK]

ELSENA



DHCP サーバーを停止させる方法としては、Vine Linux 上で「端末」を開いて shell が起動したら、 以下のコマンドを入力します。

[tori@Vine65 ~]\$ su -	
パスワード:	
<pre>[root@Vine65 ~]# /etc/init.d/dhcpd stop</pre>	
dhcpd を停止中:	[
[root@Vine65 ~]# su - tori	

※ 新たに「端末」を開いて shell が起動するたびに、DHCP サーバーが起動するので、その都度 DHCP サーバーを停止させてください。

2-1-1. Windows のネットワーク設定

- (1) Helio とホスト PC がネットワーク・ケーブルで接続されている場合はケーブルを外します。
- (2) Windows のスタート・メニューから「コントロール パネル」を開きます。
- (3) コントロール・パネルの項目から、「ネットワークと共有センター」をクリックします。
- (4) 次の図のように、「アダプタの設定の変更」をクリックします。



【図 2-1-1.1】アダプタの設定の変更

(5) 「ローカル エリア接続」があるはずですので、これを右クリックして「プロパティ」を選択します。



【図 2-1-1.2】 「ローカル エリア接続」を右クリックして「プロパティ」を選択



		[構成(C)
の接続は次の項目を	使用します(0):	1.5	
🗹 🚚 VMware Bri	dge Protocol		
🗹 🧾 VirtualBox I	Bridged Networkin	g Driver	
🗹 📕 QoS パケット	スケジューラ		
🗹 [Microsoft ネ	ットワーク用ファイル。	とプリンター共有	
✓ <u>↓ インターネット</u>	プロトコル バージョン	2 6 (TCP/IPv6)	
🗹 🗕 インターネット	プロトコル バージョ	> 4 (TCP/IP∨4)	
e la			•
0 = 1 = 0.00			
インストール(N)	自事()	0)	プロパティ(R)
説明		* ** ******	
伝送制御プロトコノ	ノインターネットプロ	トコル。相互接続	されたさまざまが
- ベットワーク暗けり用	言を提供する、以正	ミのワイト エリア 不	ットワーク フロ

【図 2-1-1.3】「インターネット プロトコル バージョン 4 (TCP/IPv4)」のプロパティを開く

(7) 通常ご使用のインターネット環境の設定に合わせて、「IP アドレス」と「サブネット マスク」を設定します。

この例では、「IP アドレスを自動的に取得する」、「DSN サーバーのアドレスを自動的に取得する」に 設定しています。固定アドレスをお使いの場合はアドレスを設定します。

設定したら、「OK」をクリックします。

ットワークでこの機能がサポートされている。 ます。サポートされていない場合は、ネット ください。	場合は、II ワーク管理	P 設定 君にji	を自動 動切な	的に取得 IP 設定を	することがで 問い合わせ
●」P アドレスを自動的に取得する(O)					
💿 次の IP アドレスを使う(S):					
IP アドレス(D		- 33	- 32	- 82	
サブネット マスク(U):		-8	- 22	82	
デフォルト ゲートウェイ(D):		-25	- 42	32 - Si	
ODNS サーバーのアドレスを自動的に耳	Q得する(E	3)			
🔘 次の DNS サーバーのアドレスを使う(I	E): —				
優先 DNS サーバー(P):	1	-83		+	
代替 DNS サーバー(A):	7	÷.	+		
□終了時に設定を検証する(L)				副総業員	設定(V)

【図 2-1-1.4】 「IP アドレス」と「サブネット マスク」の設定





2-1-2. Vine Linux のネットワーク設定

(1) Vine Linux のメニュー・バーから、「システム」⇒「設定」⇒「インターネットとネットワーク」⇒「ネ ットワーク接続」を選択します。

😼 Vine65 - VMware Player (非	営利目的の使用のみ)		
Player(P) 🕶 📙 🕶 🛃 📜	C 10		» 🗔 🚱
🌈 アプリケーション 場所	システム 🔤 💽 🧟	🚺 👘 🚺 \cdots 👼 🚥 📵	🕐 🔲 😶 🔒 💼 📢 輝 7月11
	🔜 設定 💙	三気 ユーザ向け	>
コンピュータ	システム管理 >	・ 🛋 ルック&フィール	>
toriのホーム	 ○ ヘルプ GNOME について ✓ Vine Linux について 	 インターネットとネットワーク ハードウェア システム 	は、 ホットワークプロキシ ホットワーク接続 ホットワーク接続 リモート・デスクトップ ロー ロー
2011 ゴミ箱 ◎	📄 画面のロック 실 tori のログアウト		
Vine Linux 6.5	🕑 シャットダウン		

【図 2-1-2.1】 「ネットワーク接続」を選択

(2) "ネットワーク接続" ウインドウが出たら、有線タブで 'System eth0' をハイライトして、「編集」をクリ ックします。

名前	前回の使用 🍦 追加(A)
System ethO	3分前
	デシ編集(E).
	(%) 削除(D).
	(a - manaice)

【図 2-1-2.2】 有線タブで 'System eth0' をハイライトして、「編集」をクリック



(3) "System eth0 の編集" ウインドウが出たら、IPv4 のセッティング・タブの方式で '自動(DHCP)' を 選択して、「保存」をクリックします。

System et	h0 の編集			×
続名(<u>N</u>): Sy	/stem eth0			
]自動接続する	ವ(<u>A</u>)			
与線 802.1x ·	セキュリティ IPv4	4のセッティング	ブ IPv6 のセ	ッティング
	희화 (DHCP)			
				Ľ
Addresses				
アドレス	ネットマスク	クート	ウェイ	。 追加(<u>A</u>)
				●削除(型)
DNS サーバ	(-(<u>D</u>):			
ドメインを	検索(S):			
DHCPSS	יישטיאליא ו		Walasa ay ayaa	
			1.0 24 200 i - the la	
☑ この接続	続を完了するにはI	IPv4 アドレス化	か必要になり	つます
☑ この接続	続を完了するにはI	IPv4 アドレス化	か必要になり	♪ます ト(<u>R</u>)…]
☑ この接続	続を完了するにはI	IPv4 アドレス化	か必要になり	ンます <u>ルート(R)</u> …
☑ この接続 1 すべてのユー	続を完了するには I	IPv4 アドレス化		Dます ルート(<u>R</u>)…

【図 2-1-2.3】 IPv4 のセッティング・タブの方式で '自動(DHCP)' を選択

(4) "認証" ウインドウが出たら、'root のパスワード' (この例では、toriumi)を設定して、「認証する」をクリックします。

🔍 認証	×
S	システムポリシーがシステム設定の修
E 13	この例では、 あるアプリケーションが特権の必要なアク しようとしています。このアクションを実 たのがたいます。このアクションを実 たのがたいます。このアクションを実
	root のパスワード(<u>P</u>):
▷ 詳細([2)

【図 2-1-2.4】システム設定の認証



(5) "ネットワーク接続" ウインドウで、「閉じる」をクリックします。

\Lambda ALTIMA

▼ 有線 〒 無線 📶 モ/	「イルブロードバンド 🔓 VPN 👂	DS
名前	前回の使用 🍦 追加((<u>A</u>)
System ethO	未接続	
	デン編集()	<u>E</u>)
	()親順	D)
		_/

【図 2-1-2.5】 "ネットワーク接続" ウインドウを閉じる

(6) 会社内の LAN などで、プロキシの設定があるような場合は、Vine Linux のメニュー・バーから、「シ ステム」⇒「設定」⇒「インターネットとネットワーク」⇒「ネットワーク プロキシ」を選択します。

🚭 Vine65 - VMware Player (非算	営利目的の使用のみ)		
Player(P) 🔻 📙 🖛 📜	C 🐻		» 📮 🚱
🧊 アプリケーション 場所	システム 🔤 💽 💽		📝 … 词 … 🕕 💼 🕪 👣 7月1'
	■※ 設定	>: 🛐 ユーザ向け	>
コンピュータ	🎯 システム管理	▶ 💽 ルック&フィール	>
		🛜 インターネットとネットワーク	> 📑 ネットワークプロキシ
		🔵 ハードウェア	> 🛜 ネットワーク接続
toriのホーム	Vine Linux 1501.17		> 📮 リモート・デスクトップ
The second se			
ゴミ箱	🚊 画面のロック		
	실 tori のログアウト		
Vine Linux 6.5	🕑 シャットダウン		

【図 2-1-2.6】 「ネットワーク プロキシ」を選択



(7) "ネットワーク プロキシ" ウインドウが出たら、プロキシの設定タブで 'マニュアルでプロキシの設定 を行う'を選択して、<u>Windows と同じ</u> HTTP プロキシとポートを設定します。設定が終わったら、「閉 じる」をクリックします。プロキシ設定で不明な点は社内のネットワーク管理者に確認してください。

場所:	デフォルト
プロキシの設定 無視するホスト	
○ インターネットに直接接続する	5 (<u>R</u>)
マニュアルでプロキシの設定を	を行う(<u>M</u>)
□ すべてのプロトコルで同じプロキ	⊧シを使う(<u>U</u>)
HTTP プロキシ(<u>I</u>):	ポート: 1080 🗘 詳細(
SSL プロキシ(<u>S</u>):	ポート: 🛛 🗘
FTP プロキシ(<u>E</u>):	ポート: 0 🗘
Socks ホスト(<u>O</u>):	ポート: 0 🗘
○ 自動的にプロキシの設定を行う	5(<u>A</u>)
自動設定する URL(<u>U</u>):	

【図 2-1-2.7】 "ネットワーク プロキシ" ウインドウでの設定





2-1-3. VMware のネットワーク設定

(1) VMware の "Player"をクリックし、「管理」⇒「仮想マシン設定」を選択します。

😼 Vi	ne65 - VMware Player (非営利目的の使用のみ))			
Play	er(P) 🔻 🛄 🔻 📇 📆 🍓				
	ファイル(F)	•] …]] … [7, …]] … [2] 🚺
C	パワー(P)	+			
۲	取外し可能デバイス(R)	+			
蛊	<ctrl> + <alt> + <delete> の送信(C)</delete></alt></ctrl>				
	フル スクリーン(S)	Ctrl+Alt + < <mark>Ent</mark> er>			
	ユニティ(U)				
P	管理(M)	•		VMware Tools のインストール(T)	
Γ-	ヘルプ(H)	•		メッセージ ログ(L)	
	終了(X)		61	仮想マシン設定(S) (Ctrl+D

【図 2-1-3.1】 VMware の "Player" をクリックし、「管理」⇒「仮想マシン設定」を選択

(2) "仮想マシン設定" ウインドウが出たら、ハードウェア・タブで 'ネットワーク アダプタ' をハイライトして、ネットワーク接続で「NAT」を選択します。設定が終わったら、「OK」をクリックします。

#パイス ■ メモリ ■ プロセッサ ■ ハード ディスク(SCSI) CCD/DVD(IDE) ■ スットワーク アダブタ ■ USB コントローラ ● サウンド カード ■ プリンタ ■ ディスプレイ	概要 1 GB 1 20 GB C*Temp¥Vjne621-DVD-j686.iso ファイノレ NAT あり 自動検出 自動検出	 デバイスのステータス アバイスのステータス ア 接続済み(C) アワーカン時に接続(O) ネットワーク接続 ブリッジ:物理ネットワークに直接接続(B) 物理ネットワーク接続の状態を複製(P) アダブタの設定(F) NAT:ホストの IP アドレスを共有して使用(N) ホストオンリー: ブライベートネットワークをホストと共有(H) カスタム:特定の仮想ネットワーク(U) VMnet0 (自動ブリッジ) LAN セグメント(L) LAN セグメント(S)) 詳細(V)
	追加(A) 肖邶徐(R)]

【図 2-1-3.2】 "仮想マシン設定" ウインドウで「NAT」を選択



2-1-4. インターネットへの接続確認

- (1) ネットワーク設定を変更したら、ホスト PC に外部のインターネットとネットワーク・ケーブルで接続しま す。
- (2) Vine Linux 上で Firefox ブラウザを起動します。Vine Linux のメニュー・バーから、「アプリケーション」⇒「インターネット」⇒「Fx ウェブブラウザ(firefox)」を選択(または、アイコンをクリック)します。



(3) Firefox ブラウザが起動して、インターネットに接続できれば設定成功です。



【図 2-1-4.2】 インターネットに接続できれば成功

2-2. Linux カーネルの入手

Linux 上の一般ユーザ(この例では tori)のホーム・ディレクトリ(/home/tori)から以下のコマンドを実行します。

(1) まずカーネルを入手する前に git ツリーを使ってソース・コードをビルドすることができるように、次の コマンドを実行して、Vine Linux に以下の必要なプログラムをインストールしておきます。

```
[tori@Vine65 ~]$ cd ~
[tori@Vine65 ~]$ sudo apt-get update
[tori@Vine65 ~]$ sudo apt-get dist-upgrade
[root@Vine65 ~]$ sudo apt-get install git-core texi2html chrpath
```

【注記】

社内 LAN などでプロキシがある場合は、上記の apt-get コマンドを以下のように置き換えて実行 してください(詳しくはネットワーク管理者に確認してください)。

なお、以下のコマンド列において、proxy_server と xxxxx の表記部分は、実際にご使用のネットワーク環境に応じて置き換えてください。

```
proxy_server → プロキシ・サーバー名
xxxxx → ポート番号
[tori@Vine65 ~]$ sudo http_proxy=http://proxy_server:xxxxx apt-get update
[tori@Vine65 ~]$ sudo http_proxy=http://proxy_server:xxxxx apt-get install git-core texi2html chrpath
```

 GitHub の altera-opensource サイトから linux-socfpga カーネルの git ツリーのクローンを作成しま す(数十分かかります)。

作成が完了すると、"linux-socfpga"というディレクトリができます。

[tori@Vine65 ~]\$ git clone https://github.com/altera-opensource/linux-socfpga

【注記】

error: unable to open object pack directory: /home/tori/linux-socfpga/.git/objects/pack: Too many open files

fatal: failed to read object 579a4bd2e2c0160f34d1ce8edf2a21fe12c063d1: Too many open files

のエラーが出る場合は、以下のコマンドを実行してオープンできるファイル数の制限を増やしてから、 再度 git clone コマンドを試してみてください。

```
[tori@Vine65 ~]$ ulimit -n
1024
[tori@Vine65 ~]$ su -
パスワード:
[root@Vine65 ~]# ulimit -n 65536
[root@Vine65 ~]# ulimit -n
65536
[root@Vine65 ~]# su - tori
[tori@Vine65 ~]$ ulimit -n
65536
```



(3) PC 上の Internet Explorer で、<u>https://github.com/altera-opensource/linux-socfpga</u>のアドレスを入力して、"releases"をクリックすると、アルテラ SoC FPGA 用 Linux の各種バージョン/リリースが存在することが確認できます(キャプチャ画面参照)。

ltera-opensource / linux-so	ocfpga	• Watch 37	Star 66 Fork
Code Pull requests 2	Projects 0 Insights -		
ux development repository for	socfpga	"releases"をクリック	
(7) 691,889 commits		🎎 ∞ contributors	🏘 GPL-2.0
anch: master		Fin	d file Clone or download
C Features Busines	s Explore Marketplace Pricing The	repository Lates	t commit 520eccd a day ag
Daltara anoncource / lin	uv cosfora		3 days ag
altera-opensource / IIn	их-зострда		2 days ag
<> Code Pull request	s 2 III Projects 0 Insights 🕶		
Releases Tags			
a day ago 🜼	v4.13-rc2 •• 528eccd [] zip [] tar.gz		
9 days ago 🕤	v4.13-rc1 ◆ 5771a8c		
22 days ago 🔗	v4.12 •• ef7da29 [] zip [] tar.gz		
29 days ago 🕤	v4.12-rc7 •≎ c8bc126		
on 20 Jun 👳	rel_socfpga-4.10_17.07.01_pr •© 88ee99f [] zip [] tar.gz		
on 19 Jun 🕤	v4.12-rc6 •⊙ 41f1a3e [] zip [] tar.gz		
on 16 Jun 🛇	rel_socfpga-4.10_17.06.02_pr •• 04e5253		
on 14 Jun 👳	rel_socfpga-4.1.33-ltsi_17.07.01_pr -0- 9d8d594		
on 14 Jun 👳	rel_socfpga-4.1.33-ltsi_17.06.02_pr		
on 15 Jun 😒	rel_socfpga-4.1.33-ltsi-rt_17.07.01_pr		

【図 2-2.1】GitHub の altera-opensource サイトでの linux-socfpga カーネルのページ



【図 2-2.2】 アルテラ SoC FPGA 用 Linux の各種バージョン/リリースの確認

(4) Linux カーネルの git ツリーのクローンが作成できたら、"linux-socfpga"ディレクトリに移動し、現在 のソース・ツリーを元に新たなブランチを作成します。

コミット(この例では、rel_socfpga-3.9-rel_14.02.02 を使用)からブランチ(この例では、test_0 というブ ランチ名を使用)を作成して切り替えます。

[tori@Vine65 ~]\$ cd linux-socfpga [tori@Vine65 linux-socfpga]\$ git checkout -b test_0 rel_socfpga-3.9-rel_14.02.02

ELSEN/



2-3. Linux カーネルのコンパイル

次に入手した Linux カーネルをコンパイルします。

【注記】

<u>SoC Linux</u> 道場【其ノ参】で説明した、クロス・コンパイラがインストールされて、パスが設定されていて使用できることを前提としています。

make コマンドを実行してエラーが出るようであれば、which arm-linux-gnueabihf-gcc コマン ドを実行して、クロス・コンパイラのパスが通っているか確認してください。 クロス・コンパイラのパスが正しく通っていない場合は、再度 <u>SoC Linux 道場 【其ノ参】</u>を参照し て必要な設定を行ってください。 または、次のコマンドを実行してクロス・コンパイラのパスを設定してください。

[tori@Vine65 linux-socfpga]\$ export PATH=/opt/gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabihf/bin/:\$PATH

(1) 以下のコマンドを実行して、【図 2-3.1】のような Linux カーネルのコンフィギュレーション・メニューを 立ち上げます。

[tori@Vine65 linux-socfpga]\$ make ARCH=arm socfpga_defconfig CROSS_COMPILE=arm-linux-gnueabihf-[tori@Vine65 linux-socfpga]\$ make ARCH=arm menuconfig CROSS_COMPILE=arm-linux-gnueabihf-

【注記】

make menuconfig を実行して、次のエラーが出て Linux カーネルのコンフィギュレーション画面が 出ない場合は、下記 ①、② の手順を実行してから再度 make menuconfig を実行してください。

/usr/bin/ld: scripts/kconfig/mconf.o: シンボル 'stdscr' への未定義参照です /lib64/libtinfo.so.5: error adding symbols: DSO missing from command line collect2: エラー: ld はステータス 1 で終了しました make[1]: *** [scripts/kconfig/mconf] エラー 1 make: *** [menuconfig] エラー 2

① linux-socfpga/scripts/kconfig/Makefile をエディタで開いて以下の記述を追加します。

[tori@Vine65 linux-socfpga]\$ leafpad scripts/kconfig/Makefile

HOST_EXTRACFLAGS += \$(shell \$(CONFIG_SHELL) \$(check-lxdialog) -ccflags) ¥ -DLOCALE -l/usr/include/ncurses HOST_LOADLIBES += -ltinfo -lncurses -L/usr/lib64/ この行を新規追加

② linux-socfpga/scripts/kconfig/lxdialog/dialog.h ファイルをエディタで開き次のように変更します。

[tori@Vine65 linux-socfpga]\$ leafpad scripts/kconfig/lxdialog/dialog.h

#include CURSES_LOC ↓ #include CURSES_LOC を **#include <ncurses. h> に変更**



【図 2-3.1】 カーネル・メニュー・ウィンドウ

 (2) 必要であれば、カーネルのコンフィギュレーション・メニューで、矢印キー(→/←/↓/↑)を使用して 操作し設定を行いますが、ここでは【図 2-3.2】のように単に「Exit」を選択し Enter キーを押してメニ ューを抜けます。



【図 2-3.2】 「Exit」でカーネル・メニューを終了

(3) メニューを抜けたら以下のコマンドでカーネルをコンパイルします。お使いの PC のスペックや、ネット ワーク環境にもよりますが、コンパイルには約 10~20 分程度かかります。

[tori@Vine65 linux-socfpga]\$ make ARCH=arm CROSS_COMPILE=arm-linux-gnueabihf-

ALTIMA



(4) コンパイルが正常に終了すると、arch/arm/boot の下に zImage というファイルが生成されています。 このファイルがカーネル本体になります。



Helio 用の microSD カードを ホスト PC にマウントし、microSD カード内にある zImage を入れ換え れば、生成したカーネルで動作させることができます。

これは後程、試してみることにします。

2-4. Helio に接続するためのネットワーク設定の変更

ここで再び、ホスト PC と Helio を 1 対 1 で接続する閉じたネットワーク環境に設定を戻します。

2-4-1. Windows のネットワーク設定

ALTIMA

- (1) ホスト PC がネットワーク・ケーブルでインターネットに接続されている場合はケーブルを外します。
- (2) Windows のスタート・メニューから「コントロール パネル」を開きます。
- (3) コントロール・パネルの項目から、「ネットワークと共有センター」をクリックします。
- (4) 次の図のように、「アダプタの設定の変更」をクリックします。



【図 2-4-1.1】アダプタの設定の変更

(5)「ローカル エリア接続」があるはずですので、これを右クリックして「プロパティ」を選択します。



【図 2-4-1.2】 「ローカル エリア接続」を右クリックして「プロパティ」を選択



		一構	咸(C)
	Bridge Protocol		
VirtualB	ox Bridged Networking Dr	iver	
☑ 🗐 QoS パケ	ットスケジューラ	20175	
Microsof	t ネットワーク用ファイルとプリ	ンター共有	-
	ット プロトコル バージョン 6 (TCP/IPv6)	
	ットプロトコル バージョン 4 (TCP/IPv4)	
•	m		F
インストール(ト) 肖耶余(U)	עם לי	የティ(R)
= 			

【図 2-4-1.3】「インターネット プロトコル バージョン 4 (TCP/IPv4)」のプロパティを開く

(7)「次の IP アドレスを使う」を選択して、「IP アドレス」と「サブネット マスク」を設定します。
 この例では、IP アドレスを 192.168.1.3、サブネット マスクを 255.255.255.0 に設定しています。
 設定したら「OK」をクリックします。

インターネット プロトコル バージョン	ン 4 (TCP/IPv4)のプロパティ 🛛 💽 🔜
全般	
ネットワークでこの機能がサポートされてい きます。サポートされていない場合は、ネッ てください。	いる場合は、IP 設定を自動的に取得することがで ットワーク管理者に適切な IP 設定を問い合わせ
○ IP アドレスを自動的に取得する(0))
()、)、)、(の) IP アドレスを使う(S):	
IP アドレス(D:	192 . 168 . 1 . 3
サブネット マスク(U):	255 . 255 . 255 . 0
デフォルト ゲートウェイ(D):	
🔵 DNS サーバーのアドレスを自動的(に取得する(B)
ー 💿 次の DNS サーバーのアドレスを使	ð(E):
優先 DNS サーバー(P):	
代替 DNS サーバー(A):	47 K 12
■ 終了時(ご設定を検証する(L)	【】詳細設定(V)
	OK キャンヤル

【図 2-4-1.4】 「IP アドレス」と「サブネット マスク」の設定



2-4-2. Vine Linux のネットワーク設定

(1) Vine Linux のメニュー・バーから、「システム」⇒「設定」⇒「インターネットとネットワーク」⇒「ネ ットワーク接続」を選択します。

😼 Vine65 - VMware Player ((非営利目的の使用のみ)				
Player(P) 🔻 🔛	۳			» 🗔	
🧊 アプリケーション 場	所 システム 🔁 💽 🗐		<u></u>	💵 😶 🛑 🕪 🗛 🛛 7	月1
	🔐 設定 💙	三気 ユーザ向け	>		
コンピュータ	システム管理 >	■★ ルック&フィール	>		
tori のホーム	 マントプ ♂ GNOME について グ Vine Linux について 	 マンターネットとネットワーク ハードウェア システム 	> 🐺 > 🔽 >	ネットワークプロキシ ネットワーク接続 リモート・デスクトップ]
ビデオ ゴミ箱 Vine Linux 6.5 DVD for v96	 画面のロック toriのログアウト… シャットダウン… 				

【図 2-4-2.1】 「ネットワーク接続」を選択

(2) "ネットワーク接続" ウインドウが出たら、有線タブで 'System eth0' をハイライトして、「編集」をクリ ックします。

名前	前回の使用 🎒 追加(A
System ethO	3分前
	😚 編集(E
	(😪 削除(D
	8 - marte

【図 2-4-2.2】 有線タブで 'System eth0' をハイライトして、「編集」をクリック



(3) "System eth0 の編集" ウインドウが出たら、IPv4 のセッティング・タブの方式で '手動' を選択し、 「追加」をクリックして、IP アドレス、ネットマスク、ゲートウェイ、DNS サーバーを以下のように設定し ます。設定が終わったら「保存」をクリックします。

🛜 System ethO の編集			×	
接続名(<u>N</u>): System ethO				
☑ 自動接続する(<u>A</u>)				
有線 802.1x セキュリティ	IPv4 のセッティン	グ IPv6 のセッテ	イング	
方式(<u>M</u>): 「手動				
Addresses				
アドレス ネッ	トマスク ゲー	トウェイ	🚽 追加(<u>A</u>)	
192.168.1.2 255.	255.255.0 192	.168.1.1	◎削除(<u>D</u>)	
			IP アドレス : 192.	168.1.2
DNS サーバー(<u>D</u>):	1192.168.1.1	·	ネットマスク : 255 ゲートウェイ : 192	.255.255.0 2.168.1.1
ドメインを検索(<u>S</u>):			DNS サーバー: 1	192.168.1.1
DHCP クライアント ID(E	D: [に設定する	
☑ この接続を完了する	こは IPv4 アドレス化	が必要になります	t	
		5	ν-ト(R)…	
☑ すべてのユーザーに利用	可能 🔯 キャン	/セル(<u>c</u>)	。 保存(<u>S</u>)	

【図 2-4-2.3】 IPv4 のセッティング・タブの設定内容

(4) "認証"ウインドウが出たら、'root のパスワード' (この例では、toriumi)を設定して、「認証する」をク リックします。

9. XII		×	
R	システムポリシーがシステム設定 正を阻止します。 あるアプリケーションが特権の必要なアク しようとしています。このアクションを実 root 権限としての認証が必要になります。	の 修 この例では、 root のパスワー を入力して、「	ードに toriumi 認証する」をクリック
▶ 詳細(₫	root のパスワード(<u>P</u>): ●●●●●●● 2) ② キャンセル(<u>C</u>) ② 認証	する(<u>A</u>)	





(5) "ネットワーク接続" ウインドウで、「閉じる」をクリックします。

名前	11	前回の使用	in the second se	ħΠ(Δ)
System eth0		未接続		
			S #1	集(<u>E</u>)
				余(D)

【図 2-4-2.5】 "ネットワーク接続" ウインドウを閉じる

(6) Vine Linux のメニュー・バーから、「システム」 ⇒ 「設定」 ⇒ 「インターネットとネットワーク」 ⇒ 「ネ ットワーク プロキシ」を選択します。

🚭 Vine65 - VMware Player (非算	営利目的の使用のみ)			
Player(P) 🔻 📙 🖛 📜	C 🐻			» 📮 🚱
🧬 アプリケーション 場所	システム 🔄 💽 🗐 👘		Ø	🔲 … 🕛 💼 📢 輝 7月1
	🔐 設定 💙	弐 ユーザ向け	>	
コンピュータ	ジステム管理 >	🔜 ルック&フィール	>	
		🛜 インターネットとネットワーク	>	💭 ネットワークプロキシ
A		🔵 ハードウェア	>	🛜 ネットワーク接続
tori のホーム	Wine Linux について	🝥 システム	>	🛄 リモート・デスクトップ
2				
コニ粕				
•	🍝 tori のログアウト			
Vine Linux 6.5	🕐 シャットダウン			

【図 2-4-2.6】 「ネットワーク プロキシ」を選択



(7) "ネットワーク プロキシ" ウインドウが出たら、プロキシの設定タブで 'インターネットに直接接続する' を選択して、「閉じる」をクリックします。

💭 ネットワークプロキシの設定	l	×
場所: デ	フォルト	
プロキシの設定 無視するホスト		_
() インターネットに直接接続する(<u>R</u>)		
○ マニュアルでプロキシの設定を行う	∋(<u>M</u>)	
□ すべてのプロトコルで同じプロキシを	を使う(<u>U</u>)	
HTTP プロキシ(I):	ポート: 「詳細(D)」	
SSL プロキシ(<u>S</u>):	ボート: 0	
FTP プロキシ(E):	ボート: 0	
Socks ホスト(<u>O</u>):	ボート: 0	
○ 自動的にプロキシの設定を行う(<u>A</u>))	
自動設定する URL(<u>U</u>):		
② ヘルプ(<u>H</u>)	◎場所の削除(D) 第間じる(C)	J
		_

【図 2-4-2.7】 "ネットワーク プロキシ" ウインドウでの設定





2-4-3. VMware のネットワーク設定

(1) VMware の "Player"をクリックし、「管理」⇒「仮想マシン設定」を選択します。

😼 Vi	ne65 - VMware Player (非営利目的の使用のみ))			
Play	er(P) 🔻 🛄 🔻 📇 📆 🍓		-		
]	ファイル(F)	•	10] … [] … [7 … [2 … [2	I ···] 🕕
C	パワー(P)	•			
\odot	取外し可能デバイス(R)	+			
品	<ctrl> + <alt> + <delete> の送信(C)</delete></alt></ctrl>				
	フル スクリーン(S)	Ctrl+Alt + <enter></enter>			
	ユニティ(U)				
2	管理(M)	•		VMware Tools のインストール(T)	
Γ-	ヘルプ(H)	•		メッセージ ログ(L)	
	終了(X)		6î	仮想マシン設定(S)	Ctrl+D

【図 2-4-3.1】 VMware の "Player" をクリックし、「管理」⇒「仮想マシン設定」を選択

(2) "仮想マシン設定" ウインドウが出たら、 ハードウェア・タブで 'ネットワーク アダプタ' をハイライト して、ネットワーク接続で「ブリッジ」を選択します。設定が終わったら、「OK」をクリックします。

 	概要 1 GB 1 20 GB <u>C:¥Temp¥Vine621-DVD-i686.iso ファイル</u> フリッジ(自動) あり 自動検出 あり 自動検出	デバイスのステータス ✓ 接続済み(C) ✓ パワー オン時に接続(O) ネットワーク接続 ・ リッジ: 物理ネットワークに直接接続(B) ■ 物理ネットワーク接続の状態を複製(P) アダブタの設定(F) ● NAT: ホストの IP アドレスを共有して使用(N) ● ホストオンリー: ブライベートネットワークをホストと共有(出) ● カスタム: 特定の仮想ネットワーク(U)		
	<u>這加(A)</u> 肖邶余(R)	VMnet0 (目動フリッジ) ▼ ◎ LAN セヴメント(L) LAN セヴメント(S) 詳細(V)		

【図 2-4-3.2】 "仮想マシン設定" ウインドウで「ブリッジ」を選択



2-4-4. Helio との接続確認

- (1) ネットワーク設定を変更したら、PC と Helio を 1対1 でネットワーク接続して通信できるようにします ので、下図のように PC と Helio をイーサーネット・ケーブルで接続してください。
- (2) また、USB-UART ケーブルも接続して、ホスト PC 上でターミナル・ソフト(TeraTerm など)を起動します。ボーレート「115200」、データ「8bit」、パリティ「none」、ストップビット「1bit」、フロー制御「none」に設定します。
- (3) SoC Linux 道場【其ノ壱】で作成した microSD カードを Helio のカード・スロットに取り付けます。



【図 2-4-4.1】PC と Helio の接続図(全体)

(4) Helio の電源を入れて Linux を起動してみると、もともとの microSD カードでは、カーネルの起動メッ セージとして以下のような内容になっているはずです。

Starting kernel ... Booting Linux on physical CPU 0x0 Initializing cgroup subsys cpuset Linux version 3.9.0 (jrucker@altera-linux-jr) (gcc version 4.7.3 20121106 (prerelease) (crosstool-NG linaro-1.13.1-4.7-2012.11-20121123 - Linaro GCC 2012.11)) #1 SMP Mon Nov 4 23:04:53 CST 2013

上記のログから、もともとの microSD カード内に入っている Linux カーネルのバージョンは 3.9.0 で、gcc のバージョンは 4.7.3 で、ツールチェーンは Linaro の GCC を使っていることがわかります。



2-5. コンパイルした Linux カーネルに差し換えて Helio を起動してみる

では、現在の microSD カード内の Linux のカーネル・イメージ・ファイル (zImage)を、今回コンパイルした zImage と差し換えて Helio を起動してみましょう。

2-5-1. Windows 側に zImage ファイルをコピーする

Samba サーバー経由で、Vine Linux 上にある今回コンパイルした zImage ファイルを、Windows 上にコ ピーします。このとき Helio の電源はまだ入れたまま(Linux が起動した状態)にしておいてください。

【注記】

以降の説明は、<u>SoC Linux 道場【其ノ弐】</u>で説明した、Linux マシンと Samba サーバーが既に 用意・設定されていて、使用できることを前提としています。

 Windows 上のエクスプローラのアドレス・バーに以下のように「<u>¥¥Vine65</u>」または「<u>¥¥192.168.1.2</u>」 と入力します。



【図 2-5-1.1】Windows のエクスプローラのアドレス・バーに Vine Linux のホスト名を入力

(2) ユーザ名とパスワードを入力するウィンドウが現れた場合は、smbpasswd で設定した ID (この例で は tori) とパスワード (この例では toriumi) を入力します。

Windows セキュリティ
ネットワーク パスワードの入力 次に接続するためのパスワードを入力してください: Vine621
tori ・・・・・・・ たメイン: DELL-VOSTRO に対信報を記憶する
OK キャンセル

【図 2-5-1.2】 Samba 用の ID とパスワードの入力



(3) 下図のように、Linux のファイル・システムがエクスプローラから見えれば成功です。

Samba サーバーがうまく起動すれば、エクスプローラで Linux と Windows の間で様々なファイルを やり取りすることができます。



【図 2-5-1.3】 Windows のエクスプローラから Linux 側のファイルを確認

 (4) Vine Linux 上のファイルが見えたら ¥tori¥linux-socfpga¥arch¥arm¥boot に行き、zImage ファイルを Windows 上の適当なフォルダ(Temp など)にコピーしておきます。



【図 2-5-1.4】¥tori¥linux-socfpga¥arch¥arm¥boot の下にある zImage ファイル



2-5-2. microSD カード内の zImage ファイルを差し換える

ホスト PC の SD カード・スロット(または USB カード・リーダを接続)に microSD カードを挿して、カード 内の zImage と差し換えます。

- (1) Helio の電源を落として、microSD カードを取り出します。
- (2) ホスト PC の SD カード・スロット(または USB カード・リーダを接続)に microSD カードを挿します。 下図のような表示が出たら「フォルダを開いてファイルを表示」をクリックします。

します

【図 2-5-2.1】Windows で microSD カードを開く

(3) カード内の zImage ファイルを、コピーしておいた今回コンパイルしたものに差し換えます。

念のため、もともと入っている zImage ファイルは、 zImage_org などに名前を変更しておくことをお勧めします。



【図 2-5-2.2】microSD カード内の zImage ファイルの入れ換え



(4) zImage ファイルを差し換えたら、ホスト PC から microSD カードを取り出します。



【図 2-5-2.3】 microSD カードの取り出し

2-5-3. zImage ファイルを差し換えた microSD カードで Helio を起動する

Helio のカード・スロットに microSD カードを挿して電源を入れます。

Linux カーネル起動時のターミナル表示は次のようになるはずです。

Starting kernel ...

Booting Linux on physical CPU 0x0 Initializing cgroup subsys cpuset Linux version 3.9.0-00161-ged01b8c-dirty (tori@Vine65) (gcc version 4.9.4 (Linaro GCC 4.9-2017.01)) #1 SMP Tue Jul 25 11:56:28 JST 2017

上記のログから、Linux カーネルのバージョンは 3.9.0 で、gcc のバージョンは 4.9.4、ツール・チェーン は Linaro の GCC を使っていることがわかります。

今回コンパイルした Linux のカーネル・イメージ・ファイルで起動させることができました。



3. カスタム・ドライバの作成とコンパイル(その1)

カーネルの生成が無事に終了したら、次にカスタム・ドライバを作成しコンパイルして Helio 上で動かして みましょう。

- 第2章では、外部のインターネットに接続するため <u>SoC Linux 道場【其ノ弐】</u>で設定した DHCP サ ーバーを停止させました。この章では再度 Helio と接続するので DHCP サーバーを起動させます。 Vine Linux 上で「端末」が起動している場合は、それを一旦終了させてから、再度「端末」を起動する ことで【其ノ弐】で設定した DHCP サーバーを起動させます。
- (2) Linux 上の一般ユーザ(この例では tori)のホーム・ディレクトリ(/home/tori)から以下のコマンドを実行して、Leafpad で【リスト 3.1】に示した内容の hello_driver.c ドライバ・ソース・コードを作成します。

[tori@Vine65 ~]\$ leafpad hello_driver.c

【注記】

「leafpad: ディスプレイをオープンできません:」というエラーが出る場合は、「端末」を一旦終了し再度「端末」を起動してから、上記の leafpad コマンドを再度実行してください。



【リスト 3.1】Hello ドライバ・ソース(ファイル名: hello_driver.c)の内容

作成が終了したら、Leafpad で hello_driver.c ファイルを保存して閉じます。



(3) Leafpad で【リスト 3.2】に示した内容の ドライバ用 Makefile を作成します。



【リスト 3.2】デバイス・ドライバ用 Makefile の内容

作成が終了したら、Leafpad で Makefile ファイルを保存して閉じます。

【注記】

Makefile の書式として、コマンド行の先頭はスペースではなく Tab を入力してください。 スペースではコンパイルがエラーとなるので注意してください。

(4) 以下の make コマンドで Hello ドライバをコンパイルします。hello_driver.c と Makefile を作成した ディレクトリで実行します(この例では、/home/tori)。





(5) Helio の Linux が起動したら、Helio(ターゲット)側 で以下のように root でログインして、udhcpc コマンドを実行して、DHCP サーバーから Helio に IP アドレスが付与されるようにします。

【注記】

以降の説明は、既に SoC Linux 道場【其ノ弐】で説明した DHCP サーバーが設定されていて、 使用できることを前提としています。

以降の説明では、Linux マシン (Vine Linux) の IP アドレスを 192.168.1.2 として説明しています。 IP アドレスがわからない場合は、Vine Linux 上から ifconfig eth0 コマンドを実行することで設 定されている IP アドレスを確認することができます。

socfpga login: root
root@socfpga:~# udhcpc
udhcpc (v1.20.2) started
Sending discover...
Sending select for 192.168.1.238...
Lease of 192.168.1.238 obtained, lease time 21600
/etc/udhcpc.d/50default: Adding DNS 192.168.1.1

(6) 念のため、**Helio 側** から ping コマンドを実行して、ホスト側(Vine Linux)と通信できるか確かめま す。

```
root@socfpga:~# ping 192.168.1.2

PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.

64 bytes from 192.168.1.2: icmp_req=1 ttl=64 time=1.80 ms

64 bytes from 192.168.1.2: icmp_req=2 ttl=64 time=1.35 ms

64 bytes from 192.168.1.2: icmp_req=3 ttl=64 time=1.66 ms

64 bytes from 192.168.1.2: icmp_req=4 ttl=64 time=1.35 ms

---- 192.168.1.2 ping statistics ----

4 packets transmitted, 4 received, 0% packet loss, time 3005ms

rtt min/avg/max/mdev = 1.351/1.544/1.806/0.197 ms
```

(7) Hello ドライバのカーネル・モジュール(hello_driver.ko)が生成されているはずなので、それを NFS 経由で Helio 上に運びます。ホスト(Vine Linux)側 から以下のコマンドを実行します。

この例では、ターゲット側(Helio)からホスト側(Vine Linux)のマウント先を /opt/Helio にしています。

【注記】

既に <u>SoC Linux 道場 【其ノ弐】</u>で説明した、NFS サーバーが設定されていて、使用できることを 前提としています。

[tori@Vine65 ~]\$ cp hello_driver.ko /opt/Helio/

【参考】

カーネル・モジュールとは、動的に(カーネルが動作している状態で)カーネルに組み込んで、動作 させられるプログラムのことです。一般に拡張子は ".ko" です。



(8) SoC Linux 道場【其ノ四】で説明した、busybox が一般ユーザのホーム・ディレクトリ(この例では、 /home/tori)の下にコピーされていることを前提として、以下のコマンドで生成されているファイルを確か めます。

[tori@Vine65 ~]\$ file busybox busybox: ELF 32-bit LSB executable, ARM, EABI5 version 1 (SYSV), dynamically linked, interpreter /lib/ld-linux-armhf.so.3, for GNU/Linux 2.6.31, BuildID[sha1]=de7760619a2deaccf69611887c141cff67b37dba, stripped

(9) FTP サーバーを使って busybox を Helio に運びます。以下のコマンドを Helio 側 で実行します。

【注記】

以降の説明は、既に SoC Linux 道場 【其ノ弐】 で説明した FTP サーバーが設定されていて、使用できることを前提としています。

root@socfpga:/# cd /tmp root@socfpga:/tmp# wget ftp://tori:toriumi@192.168.1.2/busybox --2013-11-05 06:13:29-- ftp://tori:*password*@192.168.1.2/busybox => 'busybox' Connecting to 192.168.1.2:21... connected. Logging in as tori ... Logged in! =>> SYST ... done. ==> PWD ... done. =>> TYPE I ... done. ==> CWD not needed. =>> SIZE busybox ... 683392 ==> PASV ... done. ==> RETR busybox ... done. Length: 683392 (667K) (unauthoritative) 100%[======>=>] 683,392 --.-K/s in 0.04s 2013-11-05 06:13:39 (17.7 MB/s) - 'busybox' saved [683392]

(10) Helio 側 で以下のように、busybox に chmod コマンドで実行権限を与えてから実行します。

<pre>root@socfpga:/tmp# chmod 755 busybox</pre>				
<pre>root@socfpga:/tmp# ./busybox ls /</pre>				
bin	home	lost+found	proc	usr
boot	init	media	sbin	var
dev	lib	mnt	sys	www
etc	linuxrc	mount_dir	tmp	

【注記】

上記の確認で、mount_dir が見当たらない場合は、次の mkdir コマンドで mount_dir ディレクト リを作成してください。

root@socfpga:/tmp# mkdir /mount_dir





(11) Helio 側 で以下の busybox のコマンドを実行して、ホスト側(Vine Linux)の /opt/Helio を /mount dir にマウントします。

root@socfpga:/tmp# ./busybox mount -t nfs -o nolock 192.168.1.2:/opt/Helio /mount_dir

(12) マウントが完了したら、以下のようにして Hello ドライバのカーネル・モジュール(hello_driver.ko)を 実行します。

	insmod コマンドで、実行中のカーネルにカーネル・モジュールを
<pre>root@socfpga:/tmp# insmod /mount_dir/hello_driver Hello Helio World!!</pre>	□ードする。これにより hello_driver 内の hello_init() 関数が 呼び出され "Hello Helio World!!" のメッセージが出力される
root@socfpga:/tmp# lsmod grep hello_driver <	lsmod コマンドで、ロードされているカーネル・モジュールをリスト 表示して、hello_driver がロードされているかを確認する
root@socfpga:/tmp# rmmod hello driver -	
Good-bye Helio World!!	・ rmmod コマンドで、実行中のカーネルからカーネル・モジュールを アンロードする。これにより hello driver 内の hello exit () 関数が
	呼び出され "Good-bye Helio World!!" のメッセージが出力される

【注記】

Helio 上で動作している microSD カード内の Linux と、Vine Linux (ホスト PC)上でビルドした Linux ソースのバージョンが異なると、上記の insmod コマンドを実行したときに、下記のようなエ ラーが出ます。

hello_driver: version magic '3.9.0-00161-ged01b8c SMP mod_unload ARMv7 p2v8 ' should be '3.9.0 SMP mod_unload ARMv7 p2v8 '

Error: could not insert module /mount_dir/hello_driver.ko: Invalid module format

このような場合は、SD カード内の Linux カーネル(zImage)を、Vine Linux(ホスト PC)上でコンパ イルして生成した zImage と差し替えてから再度 insmod コマンドを試してください。

これでカスタム・ドライバの作成に必要な環境が整い、カスタム・ドライバをターゲット上で実行できることを確認できました。

次回の SoC Linux 道場 【其ノ六】 「カスタム・ドライバの作成とコンパイル(その2)」では、 Helio 用リファレンス・デザインを入手して Helio にダウンロードします。

そして、このリファレンス・デザイン内に含まれる LED を制御するために、カスタム・ドライバを記述してコンパイルします。

作成した LED 制御カスタム・ドライバを Helio に NFS 経由で転送して実行し、ドライバによる LED ア クセスの状態を確認します。





改版履歴

Revision	年月	概要
1	2015年2月	新規作成
2	2017年8月	 Sourcery CodeBench Lite Edition for ARM GNU/Linux の配布終了に 伴い、クロス・コンパイル環境として Linaro Toolchain を使用する説明 に変更
		② ゲスト OS を Vine Linux 6.2.1 i686 から Vine Linux 6.5 x86_64 に変 更

Linux は、Linus Torvalds 氏の日本およびその他の国における登録商標または商標です。

免責およびご利用上の注意 弊社より資料を入手されましたお客様におかれましては、下記の使用上の注意を一読いただいた上でご使用ください。 1. 本資料は非売品です。許可無く転売することや無断複製することを禁じます。 2. 本資料は予告なく変更することがあります。 3. 本資料の作成には万全を期していますが、万一ご不明な点や誤り、記載漏れなどお気づきの点がありましたら、本資料を入手されました下記代理店までご一報いただければ幸いです。 株式会社マクニカ アルティマ カンパニー https://www.alt.macnica.co.jp/ 技術情報サイト アルティマ技術データベース http://www.altima.jp/members/ http://www.elsena.co.jp/ 技術情報サイト ETS https://www.elsena.co.jp/ https://www.elsena.co.jp/ 技術情報サイト ETS 4. 本資料で取り扱っている回路、技術、プログラムに関して運用した結果の影響については、責任を負いかねますのであらかじめご了承ください。 5. 本資料は製品を利用する際の補助的な資料です。製品をご使用になる際は、各メーカ発行の英語版の資料もあわせてご利用ください。